

Use of Weighted Finite State Transducers in Part of Speech Tagging

Evelyne Tzoukermann
Bell Labs, Lucent Technologies

Dragomir R. Radev
Department of Computer Science, Columbia University

Abstract

This paper addresses issues in part of speech disambiguation using finite-state transducers and presents two main contributions to the field. One of them is the use of finite-state machines for part of speech tagging. Linguistic and statistical information is represented in terms of weights on transitions in *weighted* finite-state transducers. Another contribution is the successful combination of techniques – linguistic and statistical – for word disambiguation, compounded with the notion of word classes.

1 Introduction

Finite-state machines have been extensively used in several areas of natural language processing, including computational phonology, morphology, and syntax. Nevertheless, less has been done in the area of part of speech disambiguation with finite-state transducers (Silberztein1993; Roche and Schabes1995; Chanod and Tapanainen1995).

Part of speech tagging consists of assigning to a word its disambiguated part of speech in the sentential context in which this word is used. For languages which require morphological analysis, the disambiguation is performed after the assignment of morphological tags. In this paper, we suggest two novel approaches for language modeling for part of speech tagging. The first is, in the absence of sufficient training data, to use only word classes over lexical probabilities. This claim is well demonstrated and supported in (Tzoukermann et al.1995; Tzoukermann and Radev1996). Second, we present a complete system for part-of-speech disambiguation entirely implemented within the framework of weighted finite-state transducers (Pereira et al.1994). Other works have been done using weighted finite-state transducers (FST) with a combination of linguistic and statistical techniques: (Sproat et al.1996) use weighted FSTs to segment words in Chinese, and (Sproat1995) uses them for multilingual text analysis. The system we present disambiguates unrestricted French texts with a success rate of over 96%.

2 System Overview

The input to the system is unrestricted French text; the unit over which the algorithm functions is the sentence. The system consists of a cascade of FSTs, each of them corresponding to a different stage of the disambiguation. The tagging process consists of several steps, each involving the composition of the output of the previous stage with one or more transducers. Figure 1.1 presents the main stages of disambiguation.

1. **Tokenization:** the input to the system is unprocessed French text. Each sentence is preprocessed according to several criteria of normalization,

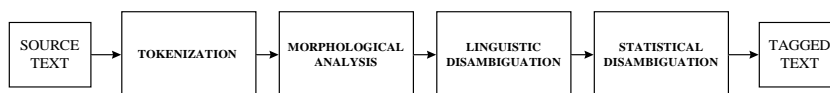


Figure 1.1: System architecture.

such as treatment of compound conjunctions as single units, treatment of uppercase words for proper names, and acronyms.

2. **Morphological analysis** is applied to the tokenized sentence; see Table 1.1, column 2. We must point out that there are over 250 tags for morphological analysis. This includes 45 verbal forms and 45 auxiliary forms, over 45 different personal pronouns, etc. These analyses were collapsed into 67 tags. We use the larger tagset mostly at the negative constraint stage, as it allows us to capture subtle agreement phenomena (see Table 1.3).¹
3. **Linguistic disambiguation**: the application of local grammars expressing negative constraints, such as noun-pronoun non agreement.
4. **Statistical disambiguation**: n-gram probabilities are computed on a training corpus and applied in terms of weights or costs on the FST transitions.

The output text consists of the disambiguated French phrase, see third column of Table 1.1 with the corresponding analyses shown in bold in the second column.

3 Weighted for Morphological Analysis

The morphological transducer is developed within the framework of finite-state morphology. The system that we have developed goes from lexical to surface form. Phonological rules are applied separately to compile verb, noun, and adjective stems. For a given verb in French, for example “venir” (*to come*), all the alternate base forms or stems necessary for the complete verb inflection are computed before the transduction from a French dictionary (Boyer1993) and stored as

¹ Note that the word “des” in Table 1.1 has three readings, namely (a) the contraction of the preposition “de” and the article “les”, (b) the partitive article, (c) the indefinite article. In the large tagset, it is represented by three distinct tags; in the shorter tagset by two tags only, i.e., the preposition tag for (a), and the article tag for (b) and (c).

Table 1.1: Morphologically tagged sentence.

Tokens	Full morphological analysis	Tags
le	pron., def. masc. sg. art.	RDM
produit	masc. sg. noun , masc. sg. past part., 3rd pers. v. pres.	NMS
liquide	sg. adj. , masc. sg. noun, 1st pers. v. ind./subj. pres., 2nd pers. v. imp, 3rd pers. v. ind./subj. pres.	JXS
qui	rel. pron. , interr. pron.	BR
entre	prep., 1st pers. v. ind./subj. pres., 2nd pers. v. imp.,	
fem bf	3rd pers. v. ind./subj. pres.	3SPI
dans	masc. pl. noun, prep.	P
le	pron., def. masc. sg. art.	RDM
processus	masc. noun	NMX
des	prep. , ind. pl. art., part. art.	P
photocopies	fem. pl. noun , 2nd pers. v. ind./subj. pres.	NFP

transitions in the list of arcs, thus forming the *arc-list* dictionary (Tzoukermann and Jacquemin1997 to appear). This approach has been described in the treatment of Spanish morphology (Tzoukermann and Liberman1990). Figure 1.2 shows the compiled base forms of the verb “venir” and some inflections associated with these stems.

The morphological FST is nondeterministic. Weights are assigned to the transitions of the FST. The lower the weight, the more likely that particular analysis will correspond to the proper disambiguation of the word. Thus, a word starting with an uppercase character will have, as a proper noun, a higher weight than the same word if it exists in the lexicon as a common noun. For example, in the sentence starting with “Marché conclu...” (*completed (or done) deal...*) the word “Marché” is tagged NPR (proper noun), NMS (masculine singular noun), and PP (past participle). In that context, it is more likely that “Marché” is a common noun rather than a proper one, thus the assignment of the lower cost to the noun form. Similarly, if a word contains only uppercase letters, it can be tagged as an acronym, even though the acronym is not present in the dictionary itself. In a similar fashion, the cost of tagging a sequence of characters as an acronym is higher than the cost of tagging the same sequence as a regular word.

Figure 1.3 shows a finite-state automaton used to tag the sequence of three words “le produit liquide”. As an example, the word “le” and the morphological tags associated with it, namely [BD3S] (3rd person singular direct pronoun), [RDM] (masculine definite article), and [UNKNOWN] are shown. At all stages of processing, we make sure that composition of finite-state transducers doesn’t fail. It happens that the source text contains typos or grammatical errors. As a result, we always allow for words to be tagged with the “unknown” tag (with a higher

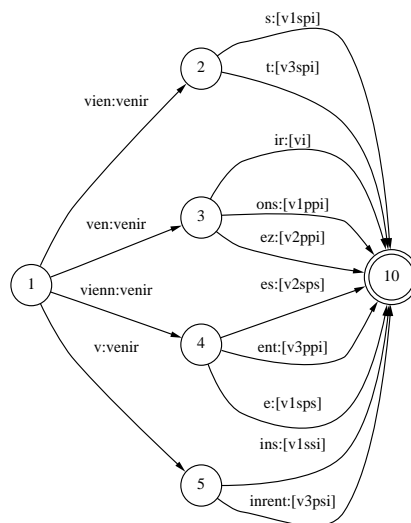


Figure 1.2: FST showing some inflections of the verb “venir” (to come).

cost) in addition to their other tags. If at the end of processing, the “unknown” tag is the only tag remaining, the system will tag the corresponding word as “unknown”. If the “unknown” tag is not the only one, it will have the highest cost of all and will not appear in the output.

Figure 1.4 shows the composition of the input string “le produit liquide” and the FST shown in Figure 1.3. One can clearly see the possible tags corresponding to the three words in the input. As negative constraints and statistical rules have not been applied yet, all weights are equal to 0 except the ones associated with the “unknown” tags.

In (Tzoukermann et al.1997 to appear), we measured the ambiguity of French words in unrestricted texts. In comparing two corpora, one of about 100,000 tokens, the other of 200,000 tokens, we found out that 56% of the words are unambiguous, 27% have two tags, 11% have three tags, and about 6% have from four to eight tags. The experiment showed three important points: a) that over half of French words are ambiguous, b) that their ambiguity varies from two tags for one fourth of the words to eight tags for the other fourth of the words, and c) that the ambiguity is constant no matter the size of the corpus.

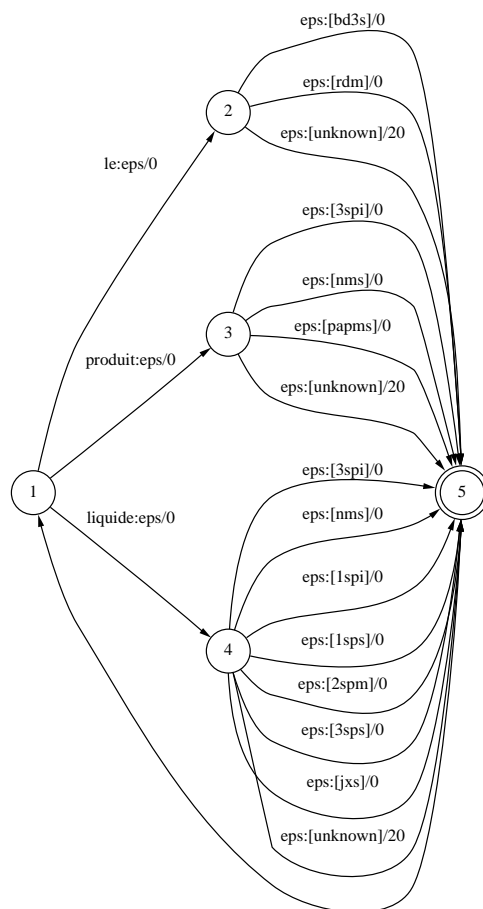


Figure 1.3: Weighted sub-FST used to tag the input string “le produit liquide”.

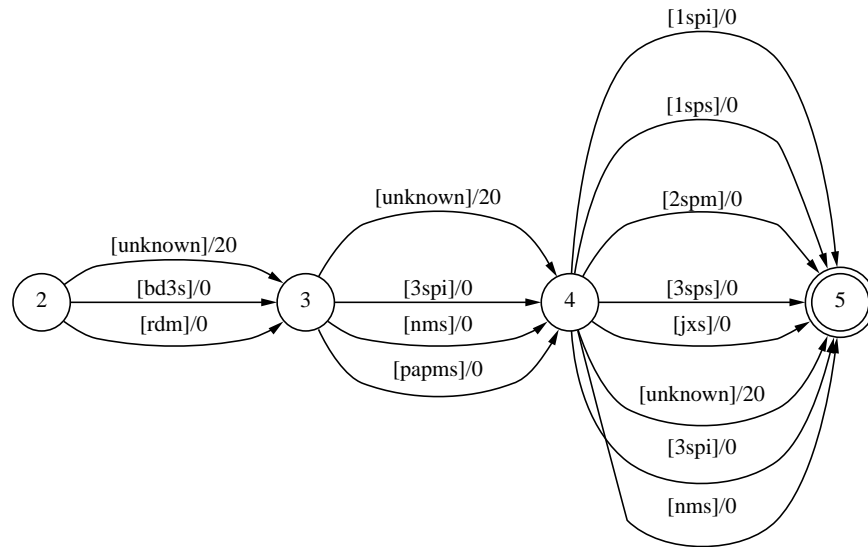


Figure 1.4: Weighted FST representing the composition of the input string “le produit liquide” and the FST shown in Figure 3

Training corpus and Genotypes

Three separate corpora were used for training². Their total size was of 76,162 manually tagged tokens³. An additional corpus of 2,200 tokens was used for testing purposes. The human tagger was given the output of the morphological analysis and had to pick the proper tag from the set. At the end of this time-consuming task, the total amount of disambiguated text was still insufficient; lexical forms of words are ignored and only their tags are considered. Table 1.2 shows the distributions of genotypes in relation to tokens and word types in the various corpora. We use the term *genotype* to capture the set of parts of speech a word can be tagged with. For example, the word “liquide” in Table 1.1 has a genotype of [JS NMS v1s v2s v3s]. As shown in Section 5, probabilities are estimated on the genotypes rather than the words (see (Tzoukermann and Radev1996) for arguments on using word class probabilities vs. lexical probabilities). Genotypes play an important role for smoothing probabilities. By paying attention to tags only and thus ignoring the words themselves, this approach handles new words that have not been seen in the training corpus. Our approach is related to Cutting *et al.* (1992), who use the notion of word equivalence or ambiguity classes to describe words belonging to the same part-of-speech categories. However, they include only words under some frequencies of occurrence, whereas our system uses word classes for every lexical item. Notice the ratio between the number of word types and the number of genotypes. In **K1** for example, there are 219 genotypes for 10,006 tokens, whereas in **K0**, 304 genotypes for 76,162 tokens, i.e., only 38% increase in the number of genotypes for a 661% raise in the corpus size.

Table 1.2: Genotype distributions from the training corpora.

Corpora	# of tokens	# of types	# of genotypes
K1	10006	2767	219
K2	34636	4714	241
K3	31520	5299	262
K0 (K1-3)	76162	10090	304

² The corpora consist of two different newspapers – one corpus was extracted from “Le Monde” newspaper (corpus of the European Community Initiative, 1989, 1990), the other from the on-line collection of French news distributed by the French Embassy in Washington D.C. between 1991 and 1994.

³ We wish to thank Prof. Anne Abeillé and Thierry Poibeau from the University of Paris for helping the manual tagging.

4 Transducers of negative constraints

Local grammars are used to represent linguistic information. This information is expressed in terms of negative constraints. These local grammars are somehow similar to the ones of (Gross1986; Mohri1994; Karlsson et al.1995), and they reflect language generalities, allowing or disallowing transitions from occurring. For example, the most common example, valid in several languages, states that an article (R) cannot precede a verb (V) as shown in the constraint R V in Table 1.3. This simple statement offers some advantages: a) in the context of the two words “le vol (*the flight*), where “le” can be either an article (*the*) or a personal pronoun (*it/him*), one can easily disambiguate “le”; if it precedes a noun (“vol”), it cannot be a pronoun, therefore it is an article. b) in the context of the two words “le manger” (*the nourishment* or *eat it*) where there is the additional ambiguity of the word “manger” (noun or verb), instead of having four readings, i.e. article-noun, article-verb, pronoun-noun, pronoun-verb, two transitions are ruled out, namely article-verb and pronoun-noun. The two remaining readings will require an additional word to disambiguate the tags in a trigram. Table 1.3 shows some examples of negative constraints. In order to favor local grammars over statistical information, negative constraints have a cost lower than n-gram genotypes obtained through statistics.

Table 1.3: Sample negative constraints.

Negative constraints	Parts of speech transitions
R V	article + verb
BR1 V2	reflexive first person pronoun + second person verb
SB BD	sentence beginning + direct object personal pronoun
W J V	numeral + adjective + verb
RDM NFS	masculine definite article + feminine singular noun

All adjacencies that have to be ruled out by the tagger can be expressed in such a way. The second rule in table 1.3 disallows the transition of a reflexive first person pronoun followed by second person verb. For instance, in the transition “me vois” (*I or you see me*) where “vois” can be first or second person, the first person is ruled out. Agreement rules are particularly well suited to be handled by this mechanism. The last transition in Table 1.3 showed how a masculine article cannot precede a feminine noun. For example, the words “le mode” (*the way* or *the fashion*) where “mode” can be either masculine or feminine singular noun, the feminine form gets ruled out to favor the masculine reading.

Stating negative rules in this manner offers an additional advantage besides rule writing simplicity. If the rule is generic for the tag, only the generic representation will be written. For instance, in the first rule of Table 1.3, R corresponds to all the articles forms, which includes 13 tags, including RD (definite article), RDP

(definite partitive article), RDMP (definite masculine plural article), RDMS (definite masculine singular article), etc. If the rule focuses on gender agreement as is the case in the last example of the table, it is possible to have a more specific tag. Figure 1.5 shows a transducer corresponding to the local grammar BR1 v2. In this particular example BR1 can be expanded into BR1P (personal pronoun reflexive 1st person plural) and BR1S (personal pronoun reflexive 1st person singular), and v2 can be expanded into 30 tags, including, among others V2PPI (verb 2nd person plural present indicative), V2SPM (verb 2nd person singular present imperative), V2SFI (verb 2nd person singular future indicative), V2SIS (verb 2nd person singular imperfect subjunctive), all the second person auxiliary forms, etc. The negative constraint transducer is used to increase the costs of certain paths in the automaton. When the output of the morphological transducer is composed with the negative constraint transducer, then the new transition costs are computed. The result is that paths including transitions that correspond to negative constraints will have an effective cost of infinity, therefore will never be selected. Since negative constraints are not allowed to be violated, costs for "unknown" tags and negative constraints were selected in such a way that paths including "unknown" tags will have smaller costs than path with negative constraints.

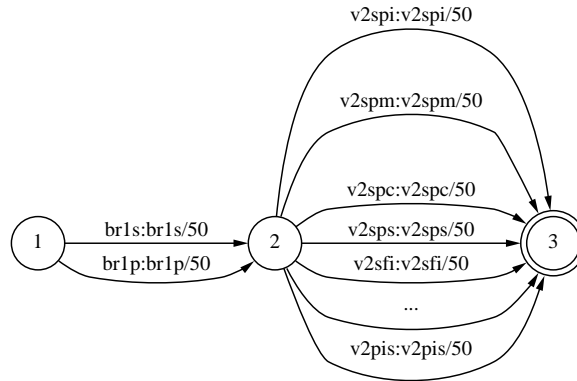


Figure 1.5: Transducer of local grammars.

A small number of constraints (in our case, only 77) can be expanded for all generic tags, thus creating a new set of 670 constraints. This was achieved using a transducer compiling rewriting rules that makes use of compositions of several transducers (Mohri and Sproat1996). This average expansion factor of 9 shows how this rule writing mechanism can be economic for the linguist.

5 Weighted FST for Statistical Tagging

We use n-grams of genotypes rather than word n-grams to estimate frequencies. Unigram, bigram, and trigram probabilities are computed from the training corpus. For example, bigram probabilities are computed by estimating the sequence of two tags, t_i and t_{i+1} , given the two genotypes, T_i and T_{i+1} , i.e., $P(t_i, t_{i+1} | T_i, T_{i+1})$, assuming that $t_i \in T_i$ and $t_{i+1} \in T_{i+1}$. For all parts of speech, the weights are derived from the frequency of a given genotype in context within the training corpus. Weights are associated with each n-gram and applied during tagging. Due to their distribution and to the disambiguation process, some words such as proper nouns, acronyms, and unknown words, are assigned higher weights.

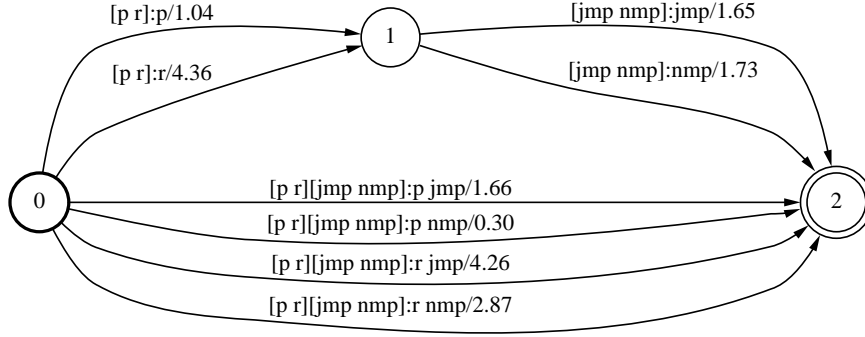


Figure 1.6: Example of a Weighted FST which tags the genotype bigram [P R] [JMP NMP]

Figure 1.6 presents a bigram genotype showing all the transitions and weights, and Table 1.4 demonstrates how weights are computed for a specific bigram and how these weights are used to make a tagging decision. The bigram [P R] [JMP NMP] occurs 141 times in the training corpus, and corresponds to the possible word “des” (*the, of the*) which has for genotype [P R] (preposition, article) and the possible word “bons” (*good ones, good*) with the genotype [JMP NMP] (masculine plural adjective, masculine plural noun). The bigram is generated automatically from the training corpus; observe in Figure 1.6 that there are 8 possible readings for the bigram (4 unigram combinations and 4 bigrams). On the one hand, the four combinations of the separate unigrams going from state 0 to 1 and from 1 to 2, each one appearing in the training corpus. In these cases, the final weights correspond to the sum of the values of [P] and [JMP], i.e. 1.66, [P] and [NMP]

with a weight of 0.30, [R] and [JMP] with a weight of 4.26, and [R] and [NMP] with a weight of 2.87. On the other hand, the sub-FST that corresponds to this bigram of genotypes will have [P R] [JMP NMP] on its input and all 4 possible taggings on its output, as illustrated in Table 1.4. Each tagging sequence has a different weight. Assume that f is the sum of all weights in a genotype bigram and f_t is the number of cases where t occurs. For all possible taggings t (in this example there are 4 possible taggings), the weight of the transition for tagging t is the negative logarithm of f_t divided by f : $-\log(f_t/f)$. Thus, the decision P JMP appears with the weight 1.66, the decision P NMP with the weight 0.30, the decision R JMP with the weight 4.26, and finally the decision R NMP with the weight 2.87. Out of these eight combinations, the lowest cost is 0.30, which means that the bigram P NMP will be selected.

Table 1.4: An example of cost computation for the bigram FST [P R] [JMP NMP].

genotype bigram	tagging	frequency	weight
[P R] [JMP NMP]	P, JMP	27/141	1.66
	P, NMP	104/141	0.30
	R, JMP	2/141	4.26
	R, NMP	8/141	2.87

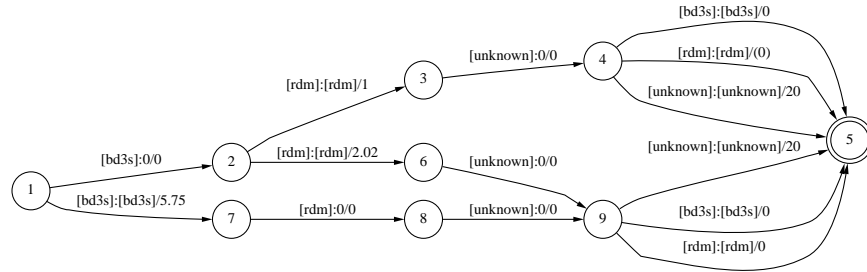


Figure 1.7: Weighted FST representing the genotype unigram [BD3S RDM] corresponding to the word “le” in the sample sentence.

6 Contextual probabilities via bigram and trigram genotypes

Using genotypes at the unigram level tends to result in overgeneralization, due to the fact that the genotype sets are too coarse. In order to increase the accuracy of part-of-speech disambiguation, we need to give priority to trigrams over bigrams, and to bigrams over unigrams.

In a way similar to decision trees, Table 1.5 shows how the use of context allows for better disambiguation of genotype. We have considered a typical ambiguous genotype [JMP NMP], corresponding to a word such as “petits” (small) which can be either masculine plural adjective (small) or masculine plural noun (small ones), which occurs 607 times in the training corpus, almost evenly distributed between the two alternative tags, JMP and NMP. As a result, if only unigram training data is used, the best candidate for that genotype would be JMP, occurring 316 out of 607 times. However, choosing JMP only gives us 52.06% accuracy. Table 1.5 clearly demonstrates that the contextual information around the genotype will bring this percentage up significantly. As an example, let us consider the 5th line of Table 1.5, where the number 17 is marked with a square. In this case, we know that the [JMP NMP] genotype has a right context consisting of the genotype [p r] (4th column, 5th line). In this case, it is no longer true that JMP is the best candidate. Instead, NMP occurs 71 out of 91 times and becomes the best candidate. Overall, for all possible left and right contexts of [JMP NMP], the guess based on both the genotype and the single left or right contexts will be correct 433 times out of 536 (or 80.78%). In a similar fashion, the three possible trigram patterns (Left, Middle, and Right) are shown in lines 18-27. They show that the performance based on trigrams is 95.90%. Disambiguation results are provided in Table 1.6. This particular example provides strong evidence of the usefulness of contextual disambiguation with genotypes. The fact that this genotype, very ambiguous as a unigram (52.06%), can be disambiguated as a noun or adjective according to context at the trigram stage with 95.90% accuracy demonstrates the strength of our approach.

Smoothing probabilities with genotypes

In the context of a small training corpus, the problem of sparse data is more serious than with a larger tagged corpus. Genotypes play an important role for smoothing probabilities. By paying attention to tags only and thus ignoring the words themselves, this approach handles new words that have not been seen in the training corpus. Table 1.7 shows how the training corpus provides coverage for n-gram genotypes that appear in the test corpus. It is interesting to notice that only 12 out of 1564 unigram genotypes (0.8%) are not covered. The training corpus covers 71.4% of the bigram genotypes that appear in the test corpus and 22.2% of the trigrams.

7 Related Research

Approaches to part of speech taggers can be divided into two types: Markov-model based taggers on the one hand (Bahl and Mercer1976; Leech et al.1983; Merialdo1994; DeRose1988; Church1989; Cutting et al.1992), and rule-based part of speech taggers (Klein and Simmons1963; Brill1992; Voutilainen1993) on

Table 1.5: Influence of context for n-gram genotype disambiguation.

n-gram	pos.	total	genotype	decision	distr.	correct	total
Unigram		607	[jmp nmp]	jmp	316	316	607
				nmp	291		
Bigram	Left	230	[jmp nmp][x]	jmp, x	71	71	102
				nmp, x	31		
			[jmp nmp][p r]	jmp, p	17	71	91
				jmp, r	3		
				nmp, p	71		
			[jmp nmp][nmp]	jmp, nmp	23	23	24
				nmp, nmp	1		
			[jmp nmp][a]	jmp, a	13	13	13
	Right	306	[p r][jmp nmp]	p, jmp	27	112	141
				p, nmp	104		
				r, jmp	2		
				r, nmp	8		
			[b r][jmp nmp]	r, jmp	22	72	94
				r, nmp	72		
			[nmp][jmp nmp]	nmp, jmp	71	71	71
Trigram	Left	32	[jmp nmp][p r][nms]	nmp, p, nms	21	21	21
			[jmp nmp][jmp nmp][x]	jmp, jmp, x	3	8	11
				nmp, jmp, x	8		
	Middle	44	[p r][jmp nmp][p r]	p, nmp, p	23	23	23
			[b r][jmp nmp][p r]	r, nmp, p	19	19	21
				r, jmp, p	2		
	Right	46	[p r][nmp][jmp nmp]	p, nmp, jmp	27	29	29
				r, nmp, jmp	2		
			[n z][p r][jmp nmp]	z, p, nmp	16	17	17
				z, r, nmp	1		

Table 1.6: Evaluation of the predictive power of contextual genotypes.

n-gram	cor.	total	accuracy
Unigram	316	607	52.06%
Bigram	433	536	80.78%
Trigram	117	122	95.90%

Table 1.7: Coverage in the training corpus of n-gram genotypes that appear in the test corpus.

	test corpus # of genotypes	training corpus # of genotypes	accuracy
1-grams	1564	1552	(99.2 %)
2-grams	1563	1116	(71.4 %)
3-grams	1562	346	(22.2 %)

the other. Even though there has been a recent surge of interest in the application of finite-state automata to NLP issues, work has only started in part of speech tagging. Roche and Schabes (1995) present a part-of-speech tagger based on finite-state transducers; they use Brill's part of speech tagger and convert the rules into finite-state transducers. Operations are accomplished on the transducers, such as the application of a Local Extension function. Transducers are converted into subsequential ones, to be deterministic. The goal of the operation is to optimize the system in terms of time and execution speed, which is crucial for a working system. The work does not focus on the disambiguation per se, but rather, on the conversion of transducers into deterministic subsequential ones.

Chanod and Tapanainen (1995; 1995a) compare two frameworks for tagging French, a statistical one, based on the Xerox tagger (Cutting et al.1992), and another based on linguistic constraints only. The constraint-based tagger is proven to have better performance than the statistical one, since rule writing is easier to handle and to control than adjusting the parameters of the statistical tagger. It is difficult to compare any kind of performance with ours since their tagset is very small, i.e. 37 tags (compared to our two tagsets of 67 and 253 tags), including a number of word-specific tags which further reduces the number of tags, and does not account for several morphological features, such as gender, number for pronouns, etc. To be properly done, the comparison would involve major changes in our system since local grammars could not be applied as is, and n-gram statistics should be re-computed. Moreover, categories that can be very ambiguous, such as coordinating conjunctions, subordinating conjunctions, relative and interrogative pronouns tend to be collapsed; consequently, the disambiguation is simplified and it is not straightforward to compare results.

8 Results and Conclusion

Using weighted FSTs to couple statistic and linguistic information has shown to be highly successful in part of speech tagging. The size of the different modules of the system is presented in Table 1.8: Our system correctly disambiguates 96% of

Table 1.8: Size of the different transducers.

	Morphology	Negative constraints	Ngram genotypes
Number of states	810,263	181	12,718
Number of arcs	914,561	39,549	2,520,846

words in unrestricted texts. We ran an experiment using 10,000 words of training corpus in order to measure the improvement of n-gram disambiguation. We tested our tagger on a 1,000-word corpus. Table 1.9 shows how the performance of the tagger improves from 92.1% using only unigrams to 96.0% using unigrams, bigrams, trigrams, and negative constraints.

Table 1.9: Tagger performance with n -gram probabilities and negative constraints.

	1-grams	1, 2 -grams	neg. cons and 1, 2, 3 -grams
10K-word corpus	92.1%	93.4%	96.0%

We demonstrated that, in the absence of more training data, the use of genotypes captures linguistic generalities about words. Additionally, genotypes are used for smoothing which seriously reduces the problem of sparse data. Bigram and trigram genotypes capture the pattern of tags in context. The system has been used in automatic indexing applications and text-to-speech system for French. In text-to-speech, words having the same orthography and a different pronunciation, can be identified via their part-of-speech. This is the case of verb/noun category where words like “président” can be pronounced either [presid Δ] (when it is a noun) or [presid(˘)] (when it is a verb), the noun/verb words such as “est” [ˌst] (noun) and [ˌ] (verb). Knowing parts of speech for text-to-speech applications also permits to compute better intonational contours. We are planning to utilize additional FST tools for local grammars so that shallow syntactic units can be studied and analyzed.

References

- Lalit R. Bahl and Robert L. Mercer. 1976. Part-of-speech assignment by a statistical decision algorithm. *IEEE International Symposium on Information Theory*, pages 88–89.
- Martin Boyer. 1993. *Dictionnaire du français*. Hydro-Québec, GNU General Public License, Québec, Canada.
- Eric Brill. 1992. A simple rule-based part of speech tagger. In *Third Conference on Applied Computational Linguistics*, Trento, Italy.
- Jean-Pierre Chanod and Pasi Tapanainen. 1995. Tagging French – comparing a statistical and a constraint-based method. In *EACL*, Dublin, Ireland. Association for Computational Linguistics - European Chapter.
- Jean-Pierre Chanod and Pasi Tapanainen. 1995a. Creating a tagset, lexicon and guesser for a French tagger. In *EACL SIGDAT Workshop*, Dublin, Ireland. Association for Computational Linguistics - European Chapter.
- Kenneth W. Church. 1989. A stochastic parts program noun phrase parser for unrestricted text. In *IEEE Proceedings of the ICASSP*, pages 695–698, Glasgow.
- Doug Cutting, Julian Kupiec, Jan Peterson, and Penelope Sibun. 1992. A practical part-of-speech tagger. Trento, Italy. Proceedings of the Third Conference on Applied Natural Language Processing.
- Stephen DeRose. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39.
- Maurice Gross. 1986. *Grammaire transformationnelle du français - I. Syntaxe du verbe*. Cantilène, 92240 Malakoff.

- Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3).
- Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Atro Antilla. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin, New York.
- Lauri Karttunen. 1983. Kimmo: A general morphological processor. In *Texas Linguistic Forum*, volume 22, pages 165–186.
- S. Klein and R. F. Simmons. 1963. A grammatical approach to grammatical tagging coding of English words. *JACM*, 10:334–347.
- Kimmo Koskenniemi. 1983. *Two-Level Morphology: a General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, University of Helsinki, Helsinki.
- Geoffrey Leech, Roger Garside, and Erik Atwell. 1983. Automatic grammatical tagging of the LOB corpus. *ICAME News*, 7:13–33.
- Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.
- Mehryar Mohri and Richard Sproat. 1996. An efficient compiler for weighted rewrite rules. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 231–238, Santa Cruz, Ca. Association for Computational Linguistics.
- Mehryar Mohri. 1994. Syntactic analysis by local grammars automata: an efficient algorithm. In *COMPLEX '94*, Budapest, Hungary. Proceedings of the International Conference on Computational Lexicography.
- Fernando Pereira, Michael Riley, and Richard Sproat. 1994. Weighted rational transductions and their application to human language processing. In *ARPA Workshop on Human Language Technology*, pages 249–254. Advanced Research Projects Agency, March 8–11.
- Emmanuel Roche and Yves Schabes. 1995. Deterministic part-of-speech tagging with finite-state transducers. *Computational Linguistics*, 21(2).
- Emmanuel Roche. 1993. *Analyse syntaxique transformationnelle du français par transducteur et lexique-grammaire*. Ph.D. thesis, Université Paris 7, Paris, France.
- Max Silberztein. 1993. *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Masson, Paris, France.
- Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for chinese. *Computational Linguistics*, 22(3).
- Pasi Tapanainen and Atro Voutilainen. 1993. Ambiguity resolution in a reductionistic parser. In *Association for Computational Linguistics - European Chapter*, pages 394–403, Utrecht, Netherlands.
- Richard Sproat. 1995. A finite-state architecture for tokenization and grapheme-to-phoneme conversion in multilingual text analysis. In *Proceedings of the ACL SIGDAT Workshop, Dublin, Ireland*. ACL.
- Evelyne Tzoukermann and Christian Jacquemin. 1997, to appear. Analyse automatique de la morphologie dérivationnelle. Lille, France. Mots possibles et mots existants.
- Evelyne Tzoukermann and Mark Y. Liberman. 1990. A finite-state morphological processor for Spanish. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki, Finland. International Conference on Computational Linguistics.
- Evelyne Tzoukermann, Dragomir R. Radev, and William A. Gale, 1997, to appear. *Tagging*

French Without Lexical Probabilities. Kluwer.

Evelyne Tzoukermann and Dragomir R. Radev. 1996. Using word class for part-of-speech disambiguation. In *Fourth Workshop on Very Large Corpora*, pages 1–13, Copenhagen, Denmark. International Conference on Computational Linguistics.

Evelyne Tzoukermann, Dragomir R. Radev, and William A. Gale. 1995. Combining linguistic knowledge and statistical learning in French part-of-speech tagging. In *EACL SIGDAT Workshop*, pages 51–57, Dublin, Ireland. Association for Computational Linguistics - European Chapter.

Atro Voutilainen. 1993. NPtool, a detector of English noun phrases. Columbus, Ohio. Proceedings of the Workshop on very large corpora.